

Algorithm Design for Networked Information Technology Systems

Springer

New York

Berlin

Heidelberg

Hong Kong

London

Milan

Paris

Tokyo

Sumit Ghosh

Algorithm Design for Networked Information Technology Systems

With a Foreword by Dr. C.V. Ramamoorthy, Professor Emeritus

With 161 Figures



Springer

Sumit Ghosh
Stevens Institute of Technology
Department of Electrical and
Computer Engineering
Castle Point on Hudson
Burchard Building, Room 211
Hoboken, NJ 07030
USA

Cover illustration: ADDM algorithms encapsulate a fundamental microcosmic design principle that governs the organization of complex, real-world phenomena: astronomical systems, world civilization, computer electronics, and biological systems.

Library of Congress Cataloging-in-Publication Data

Ghosh, Sumit, 1958–

Algorithm design for networked information technology systems / Sumit Ghosh
p. cm.

Includes bibliographical references and index.

ISBN 0-387-95544-5 (hc: alk. paper)

1. Computer algorithms. 2. Data structures (Computer science) 3. Information networks.

I. Title

QA76.9.A43G46 2003

005.1—dc21

200344588

ISBN 0-387-95544-5

Printed on acid-free paper.

© 2004 Springer-Verlag New York, Inc.

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY 10010, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

SPIN 10886377

www.springer-ny.com

Springer-Verlag New York Berlin Heidelberg

A member of BertelsmannsSpringer Science+Business Media GmbH

In Memory of My Loving Parents

Foreword

I felt deeply honored when Professor Sumit Ghosh asked me to write the foreword to his book with an extraordinary perspective. I have long admired him, first as a student leader at Stanford, where he initiated the first IEEE Computer Society's student chapter, and later as an esteemed and inspiring friend whose transdisciplinary research broadened and enhanced the horizons of practitioners of computer science and engineering, including my own. His ideas, which are derived from his profound vision, deep critical thinking, and personal intuition, reach from information technology to bioscience, as exhibited in this excellent book. To me, an ordinary engineer, it opens up a panoramic view of the Universe of Knowledge that keeps expanding and inspiring, like the good Indian proverb, which says, "a good book informs you, an excellent book teaches you, and a great book changes you." I sincerely believe that Professor Ghosh's book will help us change and advance the methods of systems engineering and technology.

Vision

Inspired vision sees ahead of others what will or may come to be, a vivid, imagined concept or anticipation. An inspired vision personifies what is good and what like-minded individuals hope for. Our vision is one of creating an Internet of minds, where minds are Web sites or knowledge centers, which create, store, and radiate knowledge through interaction with other minds connected by a universal shared network.

This vision will not just hasten the death of distance, but will also incarcerate ignorance. We note that the rapid growth of many disciplines in the sciences, humanities, technology, and business, has been promoted by the convergence of ideas through knowledge transfer across traditional boundaries. This is what a distinguished business leader and entrepreneur, Professor George Kozmetsky, calls the transdisciplinary movement in education. Professor Ghosh is a great mentor, who teaches by this principle. In some sense,

knowledge transfer across traditional disciplinary boundaries is the first phase in the Internet of the Mind paradigm.

Preliminary Issue

A critical and difficult to resolve issue is how to fine tune educational processes and maximize the flow of educational information from the source—the transmitter (knowledge bases, mentors, books, teachers, etc.)—to the needy individual, the student at the receiving end. These processes must ensure that the recipient obtains the needed education in the most effective way at the most appropriate time and place. Two issues are paramount here: The first is how to make knowledge “fluid” ; the second is how to personalize educational processes to fit the intellectual infrastructure of the recipient. The concept of fluidity is similar to the Keynesian concept of liquidity of capital, which makes knowledge easily storable, transferable, and transportable across heterogeneous intellectual boundaries. The second issue will be addressed later.

Knowledge and Technology: Their Growth and Implications

We shall now tread on the unknown and unexplored area of knowledge and technology growth. We are fully aware of their exponential growth rates, but unfortunately, there are no precise methods or rigorous approaches to measure them. A Harvard Business School study many years ago found that “knowledge doubles every seven years.” More recently, former U.S. President Bill Clinton and Vice President Al Gore have indicated in their speeches that knowledge doubles every three years, and they refer to U.S. Department of Education studies.

The technology growth rate can be surmised from the progress of the high-tech semiconductor and communication industries. The famous Moore’s law states that semiconductor chip performance doubles every eighteen months; while the price stays the same. Thus, we shall assume that the knowledge growth rate based on “expert” evaluation is about 30% per year; likewise, the technology growth rate based on Moore’s law is about 66% per year. In a later section, we shall try to answer an apparent anomaly: Why is the growth rate of technology (which is derived from knowledge) much higher than the growth rate of knowledge itself?

Knowledge Obsolescence Rates

Bill Joy, the chief scientist of Sun Microsystems, has surmised in his featured interviews and articles that professional obsolescence creeps at the rate of 20% per year. In other words, after five years, we will be strangers in our professional disciplines, no better than college freshmen. One may quibble about the magnitude of the rate, but the trend appears to have some validity. This

obsolescence figure may be more applicable to mature professionals beyond the age of 40 who have not actively kept up with technological advances than to younger ones.

Technology Growth, Knowledge Growth, and Moore's Law

Technology is derived from scientific knowledge, but Moore's law tells us that technology grows much faster than knowledge. We shall offer a subtle explanation. Knowledge is a result of rational and physical discoveries. These processes can be slow and incremental, but they rarely leapfrog. Knowledge and its discoveries are like dots on the radar screen in an air traffic control center: New dots appearing, old dots always moving and changing. Technology is derived from connecting and linking the dots and creating useful patterns that can be made from them. It is like knitting a beautiful mosaic that connects and provides meaning (satisfies human needs) to the patterns. Given N dots (e.g., knowledge discoveries or inventions), the number of patterns they create is an exponential power of N . This heuristically explains Moore's law and the more rapid growth of high-tech industry. The dots and patterns are always moving and changing, and this also is a prominent characteristic of technology. Technology always advances faster than knowledge. The former subsidizes its own growth and that of science through the creation of automated tools and computer-aided adjuncts. This, in turn, fuels the accelerated growth of technology, à la Otto Hahn's chain reactions and the Livermore loops.

The Challenge

Our real challenge: how to develop a feasible and realizable approach to reducing the nonlinearly growing chasm between the advancement of technology and the increase in the knowledge content of any individual. In theory, the chasm cannot be completely and conclusively bridged. Although this raises many unanswered—and perhaps unanswerable—questions, several approaches that can slow down the overtake and obsolescence are conceivable. A key approach is compression of learning time by innovative teaching methods, which underscores the need to personalize educational processes to fit the intellectual infrastructure of the recipient. In this book, Professor Ghosh proposes interdisciplinary approach as the fundamental source of new teaching techniques. This transdisciplinary approach is fluid-like and flexible, allowing one to easily penetrate a number of different disciplines at once. The objective of a proper match between the knowledge generators and knowledge recipients is analogous to the issue of impedance matching in electrical engineering, which ensures an efficient transfer of electric power from the power-generating stations to the power-consuming towns. The objective is also critical because it holds the key to opening a path for every willing individual to continue to be an active and valuable participant in society, despite the march of time. We

are now living longer and healthier lives and working for many more years, and the idea of mandatory retirement has clearly fallen from grace. This then propels us to envision an environment and an educational framework in which a new knowledge infrastructure may be established in each individual to help him or her remain matched with the advancing technology. It then becomes the purpose of the environment and educational framework to provide caring and personalized mentors to help in the assimilation of new technologies; casting them in the molds of individuals' infrastructures for ultrafast absorption. Even as each of us ages with time, the promise that we can continue to keep pace with exciting new technologies and remain valuable and contributing members of society as long as we wish is a shining beacon of hope.

The Three Visions and Their Manifestations in This Book

Through many decades of my association with academic research and teaching, I have come to a firm belief that the interplay of three types of vision is critical for progress. These include (1) eagle vision, (2) frog vision, and (3) fish vision. The eagle soars high, has a commanding view of the entire ground in every direction, and, when necessary, is able to bring its eyes to a pinpoint focus on a scurrying rat, miles below. In this book, the eagle-eye vision constitutes the overview, a survey, a perception about the integrated whole, from the past, the present, and the future. Frog vision refers to microscopic vision, or an up-close view of an object that reveals an extraordinary level of detail. Here, the frog-eye vision is manifested in the details of the algorithms, methods, and concepts that the book articulates so well. The fish views the entire world outside the water, all 180 degrees, compressed through a narrow cone of 98 degrees. However, its vision can detect fast changes because its eyes' fields of vision do not overlap; and the fish cannot afford to shut its eyes even for a moment. The fish-eye vision assumes the form of an emphasis on the dynamics of change and movement and the inevitable progress and advances that the book will produce.

This book is not just for specialized scientists and engineers; its essence is easily accessible to the nonspecialists who will find their curiosity aroused, their need to learn stimulated, and their minds changed.

Dr. C.V. Ramamoorthy, Professor Emeritus
EECS Department, University of California, Berkeley
April 3, 2003

Preface

Overview

Networked information technology (NIT) systems, also referred to as network-centric or net-centric systems, are already pervasive and promise to become even more encompassing, powerful, highly capable, and greatly useful in the future, touching nearly every element of human endeavor and society. Already, today, NIT systems underlie much of e-commerce, banking and financial transactions, online airline reservation systems, worldwide communications, distance learning, and remote data acquisition and control, to name a few. In the near future, NIT systems will encompass patient medical records, remote surgery, intelligent transportation systems (ITS), communications and control in deep space, and innumerable innovative services and products, limited only by our imagination. Under the Future Combat Systems (FCS) program, the U.S. military, especially the U.S. Army, is committed to a total transformation to bring itself in line with the new network-centric warfare idea. Under the emerging Homeland Defense program, the objective is to preempt threats and attacks on the nation through processing a tremendous volume of intelligence data from a large number of diverse sources, without becoming overwhelmed from information overload, and coordinating the activities of geographically dispersed public safety agencies in a precise, accurate, and timely manner. Fundamentally, in an NIT system, first, the relevant data are identified and acquired from geographically dispersed points; second, the raw data are transformed into usable information through computational intelligence techniques; and third, the information is disseminated to other geographically dispersed regions, as needed. The traditional, uniprocessor-based, centralized algorithms cannot represent NIT systems faithfully and are, clearly, inadequate and unnatural. Synchronous distributed algorithms are equally inappropriate because they represent a simple extension of the centralized approach. Asynchronous, distributed decision-making (ADDM) algorithms constitute the most logical approach to serve as the underlying control of NIT systems and is the subject of this monograph.

NIT systems are complex, as opposed to toy academic problems, and are intended to operate in the real-world environment. If the incredible potential of the information age is to be brought to fruition, NIT systems must be designed precisely and operated correctly. This, in turn, requires the development of a new scientific and pragmatic approach. When designed correctly and effectively, NIT systems may last a very long time, perhaps 50 to 150 years. The consequences of failure may be serious, ranging from unnecessary loss of lives to a collapse of the public's confidence. As an example, consider VHDL, a very high-speed integrated circuit hardware description language, that was intended to be a state-of-the-art tool to describe and validate any electronic design accurately and efficiently. Initiated in 1985, a billion dollars has already been invested in the VHDL effort by the government and industry. However, its most basic objectives—accuracy and efficiency—that were mandated in its specifications back in 1985, are far from being realized. More important, its limitations have done little to facilitate the growth of innovative and complex design capability. The principal cause underlying the inadequacies of VHDL may be traced to the incorrect application of the notion of timing to hardware systems that are intrinsically asynchronous and distributed.

As a second example, consider the following occurrence in New Jersey involving the E-ZPass system [1], a cashless toll collection system for interstate highways. Motorists are required to purchase magnetic toll cards of specific dollar values and place them on their visor. When passing through the toll booth at a maximum speed of 15 mph, the toll is magnetically deducted from the card and the driver passes through the toll booth. For a period of time following the installation of E-ZPass, motorists received in the mail citations from the New Jersey Department of Motor Vehicles for speeding along the toll highways in excess of 80 to 150 mph. Apparently the E-ZPass system was being used to record the times a vehicle passed through the toll booths, from which the elapsed time between two successive toll booths was calculated and the car's speed computed. The only problem was that many of the cited drivers were driving older vehicles that were not even capable of running faster than 65 mph. Protests by the cited drivers were initially met by stiff official resistance, but when an investigation was eventually launched, it was revealed that the clocks in the different toll booths in the E-ZPass system were never synchronized. The architects responsible for computing the vehicle speeds had overlooked the basic notions of timing and clock drifts in distributed systems.

For a third example, consider the worldwide Internet whose spectacular growth and success are rapidly being dwarfed by its lack of security, underscored by its fundamental inability to track IP packets and prevent denial-of-service attacks. Triggered by the events of September 11, 2001, fueled by the belief that the Internet may have been used by the terrorists for money laundering, and threatened by the fear that dramatic cyberattacks are not preventable, increasingly, courts and corporations are attempting to wall off portions of cyberspace. In so doing, according to Lessig [2], they are destroying

the Internet's potential to foster democracy and economic growth worldwide and ending the Internet revolution just as surprisingly as it began.

Relative to the Internet and in the light of the growing attacks on the Internet from computer viruses worldwide, even reputed scientists and corporate executives of leading NIT companies have acknowledged, publicly and privately, that the Internet is far too complex to secure. At the NSF-sponsored workshop on the Modeling and Simulation of Ultra-Large Networks: Challenges and New Research Directions, held at the Sheraton in Tucson on November 19-20, 2001, the consensus among thirty top-flight researchers from the areas of network design, simulation modeling and analysis of networks, and modeling and simulation methodology was that the key principles underlying the successful scaling of the Internet are neither known nor understood. For the sake of science and our progress towards the future, respectfully, this is not acceptable. We conceived and designed the Internet and we must understand it. Lawson [3] traces the origin of the complexity in the computer industry to the deployment of compromised hardware design for all types of computing, leading to the demand for unprecedentedly complex system software that, despite involving thousands of code developers, was never completely understood. Leading researchers from the top telecommunications companies [4] have expressed their serious concern over the occurrence of inconsistencies and failures in the context of feature interactions and the current inability to understand and reason about these events. For example, while private telephone numbers are successfully blocked from appearing on destination caller ID screens under normal operation, as they should be, these private numbers are often unwittingly revealed during toll-free calls [5]. As a second example, despite the telephone subscriber paying a monthly fee for the caller ID service, incoming phone calls from the outside, including those initiated by the local telephone service provider, often show up as out of area on the consumer's caller ID display. Under such difficult circumstances, the scientific tradition recommends that we not be overwhelmed, but rather search for the fundamental principles that are usually few in number, easy to comprehend, and on which all systems, regardless of their complexity, are based. The goal of this book is to uncover these fundamental principles, where possible, in the form of ADDM algorithms. The vision underlying this book finds support from four recent discoveries.

The first is our proven ability to successfully design and operate digital computers consisting of millions of logic gates. Despite the bewildering number of gates, the key principles underlying the success are (i) there are only a handful of different types of gates, (ii) each gate is a self-contained entity [6] whose behavior and interactions with other gates are simple and well understood, and (iii) the gates are organized hierarchically to successively build bigger and more complex building blocks. The second is the discovery that underlying the perplexingly complex behavior of an ant colony, consisting of hundreds of thousands to millions of ants, are only a few types of ants—queens, foragers, guards, etc.—and that every ant within any given type is

guided by a few simple rules. The third is the discovery by geneticists that, despite the overwhelming number of genetic combinations that are conceivable in the millions of lifeforms, Nature employs only a limited number of genes, repeatedly using them over and over to create a seamless ocean of different lifeforms. The fourth is the growing acceptance among geologists that behind all the complex and bewildering physical changes we see at the earth's surface, is an underlying simple set of principles, the key being the mantle's massive heat engine that drives the tectonic plates. The inspiration from these discoveries is that underlying the highly complex yet successful systems may be a set of constituent elements that are well defined and interact through simple, provably correct rules.

The motivation for precision in NIT system design comes from another most unusual and unexpected source, namely, law enforcement. The rapid proliferation of NIT systems in society and the growing concern over how to prosecute increasingly sophisticated NIT-related crimes, coupled with the socially accepted, strict, legal threshold implied in beyond reasonable doubt, virtually necessitates precision in every phase of NIT systems design. Conceivably, spurred by law enforcement, legislation may enact laws mandating precision in NIT systems.

Careful analysis of the VHDL and E-ZPass examples and other difficulties in the distributed systems literature points to deeper problems. In the article titled, "Parallel Processors were the Future . . . and May Yet Be," in IEEE Computer (December 1996), Michael Flynn of Electrical Engineering at Stanford University observes candidly that the promise of parallel processing has not been fulfilled. He notes two primary reasons. First, for most science, the mathematical representations were developed largely in the nineteenth century, and they represent a simplified idealization of reality. Second, and more important, the mathematical modeling reflects the human reasoning process as sequential. Flynn's observation is true, as evidenced by the fact that although most real-world, large-scale, computing-based systems today are driven by centralized algorithms, the literature in electrical and computer engineering, computer science, and operations research is dominated by centralized algorithms and synchronous distributed algorithms. As stated earlier, the latter constitutes a logical extension of the centralized approach and inherits the same fundamental limitations. Examples of the use of centralized and synchronous distributed algorithms may be found in payment processing, airline reservation systems, inventory management, traffic control, etc. Flynn's two reasons may be complemented by a third reason, namely, our desire to find a simple parallel approach that applies to all problems, quickly and uniformly.

The real world, however, is complex and much like fractals. The deeper we dive and the greater our desire to understand the fundamentals in depth, the more complex it gets. It is, therefore, no surprise that the desire to uncover a simple parallel approach that applies to all problems has not yet been realized. In truth, if such a simplistic approach did exist, life for us scientists would

be boring, uneventful, and uninteresting. There would be no room left for challenges or creativity.

Anil Nerode of the Mathematics Department at Cornell University observes that among traditional engineering and computing research personnel, given a real-world problem, there is a strong tendency toward quickly establishing a mathematical model. Because the model is subject to the available analytical manipulation tools, a belief develops that the real-world problem has been successfully rendered amenable to rigorous mathematics. Nerode points out that this belief may be misplaced in that the very process of deriving a quick mathematical model strips the real-world problem of its essential characteristics. Thus, the mathematical model is grossly approximate. In contrast, an effort to develop a physical model of the problem may be far more challenging, but it is likely to yield highly accurate and realistic results.

Aim and Scope

This book presents a radically different paradigm, one where the basic premise is to precisely understand and accurately model real-world NIT systems, in their true form, to the best of our current ability. The paradigm begins with the recognition that many of the classical mathematical representations and models that we have grown up with since the nineteenth century may not apply meaningfully to the complex NIT systems of today. Practical examples include computer networks such as the Internet, banking infrastructure, credit card transaction processing networks, and inventory management systems. Second, the lens of the sequential human reasoning process through which we learned to view reality may not always provide the correct view of the world.

The author's pursuit of key problems from a number of disciplines ranging from hardware description languages, networking, inventory management, military command and control, and banking to intelligent transportation systems, and the synthesis of innovative control algorithms has serendipitously revealed this unique insight: that there is a fundamental micro-cosmic design principle of this universe, an archetype that extends from the astronomical-sized galaxies down to the minutest atom in the universe. At any level of abstraction, the subsystems of a system inherently possess independence, autonomy, and their own sense of timing; where they interact with each other, the interaction is asynchronous. The independence of each subsystem with respect to all others poses no paradox. It refers to the fact that the existence of a subsystem is independent of all others. Thus, even when a star undergoes a supernova, the rest of the universe continues to exist, although it may be subject to the flying debris and radiation from the dying star. Furthermore, at any level of abstraction, while the behavior of the subsystems and the interactions between them may be encapsulated through relatively simple rules, the repeated application of these rules to increasing numbers of constituent elements gives rise to systems of increasing complexity. In the discipline of biology, according to Franklin Harold [7], every cell constitutes a

unitary whole, a unit of life, and a tremendous diversity of complex lifeforms results from the interaction of a number, say N , of cells that come together, where N may range from two to millions, or more. Although there may be data and information dependency between two or more subsystems of a system, each subsystem possesses its own processing or decision-making engine. The interactions enable the subsystems to learn, grow, and evolve.

This book develops and presents ADDM algorithms as an encapsulation of the universal micro-cosmic design principle, serving in the role of the underlying control and coordination of NIT systems. It develops ADDM algorithms as a systematic, scientific, and canonical approach to the design and analysis of NIT systems. In essence, however, ADDM algorithms may constitute the scientific core of any real-world system. An ADDM algorithm that is successfully designed for a given NIT system reflects the higher meta-level purpose or intent of the system while underlying the behavior of every constituent subsystem. Metaphorically, the ADDM algorithm represents the big picture, the bird's-eye view of the total NIT system behavior. Prof. C.V. Ramamoorthy of the University of California, Berkeley, refers to this as eagle vision. In the classic Public Broadcasting Service (PBS) television interview with Bill Moyers titled, "Joseph Campbell and the Power of Myth," Joseph Campbell rephrases the philosopher Schopenhauer: In this universe, everything [including every human life] influences everything else as if all lives are the dream of a single dreamer in which all the characters are also dreaming. Campbell describes the idea of the net of gems, also known as the net of Indra, the king of gods, as one where every life is represented as a gem, every gem reflects all others, and all gems are rising, spontaneously, simultaneously, and in harmony, as if there is an intention of cosmic proportion behind it. Fundamentally, the goal of ADDM algorithms is to encapsulate the intent underlying any complex real-world system in a systematic scientific manner. ADDM algorithm design represents an integrated exercise in both top-down design, where one starts with an intended overall behavior and attempts to synthesize the behaviors of the constituent lower-level entities, and a closely knit bottom-up phase, where the focus is on the design of the traits of the individual entities, such that together they yield the intended high-level behavior. The ADDM algorithm design effort is generally accompanied by a validation effort—a testing of the theory under realistic conditions before developing and deploying the actual system—through modeling and asynchronous distributed simulation of the underlying NIT system under ADDM algorithm control.

The content of this book is based on actual research and the experience of the author and his research group over the past twenty-two years, acquired through conceiving, building, testing, and analyzing real-world systems across a number of diverse disciplines. This book stands apart from virtually all of the books currently available on distributed algorithms in that it views asynchronous distributed algorithms in their true natural form. No constituent entity is presumed to possess any kind of global knowledge. In contrast, most of the currently available books cast distributed algorithms in the light of

centralized thinking and frequently refer to concepts such as shared variables, global snapshots, knowledge of every state of a system at every time instant, awareness of deadlock in the entire system, asynchronously connected processors maintaining communication with a central facility, and certainty of distributed termination, all of which fundamentally contradict the essence of asynchronous, distributed, real-world systems. A likely reason may be that the authors of these books analyze the problems, understandably, through sequential reasoning and then inadvertently presume that the constituent entities have access to the authors' global knowledge of the problems. The authors of several books reveal a keen underlying desire to bring inherently asynchronous real-world processes under synchronous control, arguing that synchronous algorithms are simple to design. In one book in particular (*Parallel and Distributed Computation: Numerical Methods*, by Dimitri P. Bertsekas and John N. Tsitsikilis, ISBN 0-13-648700-9, 1989), the authors write, "it is quite hard to synchronize a data communication network and, even if this were feasible, it is questionable if the associated overhead can be justified." Clearly, the fundamental truth that geographically dispersed, inherently asynchronous systems may never be accurately encapsulated by synchronous algorithms and global synchronization is lacking.

Organization and Features

Chapter 1 defines NIT systems and ADDM algorithms and explains their relationship following a review of the current distributed systems literature and a critical analysis of the current computing paradigms. Chapter 2 presents the nature and fundamental characteristics of ADDM algorithms in light of how they influence and relate to each other and together solve the challenges underlying the NIT system. These principles are neither esoteric nor meant for toy academic problems. They are simple yet canonical and apply to actual problems in the real world. The U.S. Defense Advanced Research Projects Agency (DARPA) [8] observes that today's complex systems pose a formidable challenge in the form of scalability of the underlying control algorithm. A solid understanding of the principles, presented in Chapter 2, is the only known scientific approach today; by applying them, one may synthesize precise ADDM algorithms for NIT problems that will function correctly if the number of constituent entities were to increase from 10 to 50 or 5,000 or even 100,000, subject to other design limitations.

Chapter 3 presents a series of actual case studies in which ADDM algorithms have been successfully synthesized for a number of NIT systems, implemented through large-scale distributed simulations, and subject to performance analysis. The scope of the problems ranges from distributed discrete-event simulation of hardware description models in VHDL, railway networks, international payment processing, military command and control, fault simulation of digital systems, domestic payment processing in a partially connected network of banks, and a hierarchical distributed dynamic approach to

inventory management. As evidence of the real-world nature of the problems studied, consider the following anecdote. The U.S. Department of Transportation (DoT) had recently commissioned [9] the Institute for Simulation at the University of Central Florida to review all of the commercial simulators manufactured in the United States and abroad and the MIT-developed simulator for the Boston Tunnel project, to examine their suitability for autonomous simulation of a large number of drivers and vehicles. The Institute evaluated a total of 51 simulators and provided a short list of 11 simulators that they felt met the DoT's criteria. After closer examination, the DoT concluded that neither of the commercial simulators in the short list nor the MIT-developed simulator satisfied its needs. In contrast, the DICA algorithm [10] [11] developed by the author's research group has been successfully validated through an asynchronous distributed simulator executing on a network of more than 65 Sun spare 10 workstations, for a total of 45,000 autonomous vehicles representing a highly congested traffic scenario in Rhode Island.

Chapter 4 details key issues in debugging complex NIT systems. Chapter 5 argues the crucial importance of proofs of correctness for ADDM algorithms and presents techniques to develop such proofs. Chapter 6 presents a mathematical framework for synthesizing ADDM algorithms, starting with a centralized algorithm, and illustrates it for a specific military command and control problem. Chapter 7 reviews the conventional approach to performance analysis of distributed systems and presents a radically different paradigm for ADDM algorithms. Chapter 8 argues the need to perform perturbation analysis. It presents key principles and reasoning and describes results obtained from the stability analysis of representative ADDM algorithms. Chapter 9 recognizes the need for creativity and imagination in the synthesis of ADDM algorithms, argues from logical principles the role that high-quality interdisciplinary thinking may play in realizing this goal, and presents new techniques to instill such thinking in future NIT engineers and scientists. Finally, Chapter 10 summarizes the book and presents some reflections on the benefit and impact of ADDM algorithms in the future.

Audience

This book has been developed with three types of audiences in mind and has been written to facilitate straight-to-the-point, intense, and in-depth self-study. The only prerequisites are knowledge of the basic principles of physics and mathematics, logic, common sense, and a serious desire to learn. First, this monograph is a reference book intended to serve as a reader for a graduate or senior-level undergraduate course titled, "Algorithm Design for NIT Systems" within the traditional computer engineering program or a new program such as the one in Networked Information Systems Engineering currently being developed at Stevens Institute of Technology. The monograph's intent is to constitute a key starting point for developing a theoretical and practical understanding of NIT systems in doctoral, graduate, and advanced undergrad-

uate students who plan to pursue a career in NIT system design. It presents a scientific and pragmatic approach, developed in a canonical fashion, focusing on the underlying physical meaning and the fundamental mathematical reasoning, unlike the tradition of a purely mathematical treatment that is often devoid of pragmatics. The author hopes that the book will not only help students learn how to design, operate, maintain, and evolve NIT systems precisely but also inspire them to take on the role of leading thinkers and guide the evolution of the information age. In addition, the newly created interdisciplinary programs in NIT and business management, such as the Howe School of Technology Management at Stevens Institute of Technology (Hoboken, NJ), the Indian Institute of Information Technology and Management (Gwalior, India), and the Jerome Fisher Program in Management and Technology at the University of Pennsylvania (Philadelphia, PA), may also use this reference book to teach the technology of ADDM algorithms within the context of a high-level overarching course in future technologies. Second, the book targets policy makers in NIT within the industry, military, and government and aims to educate them from a fundamental perspective, in an intensive short-course style, on the nature, scope, potential, and limits of NIT systems. Third, the monograph offers to managers, consultants, and other professionals of the NIT industry successful examples of NIT system design, analysis of their performance, and the design of unique measures of performance. To practitioners, the book presents an integrated approach to the issues of algorithm design, distributed systems, and concurrent processing.

Acknowledgments

The author gratefully acknowledges the insights, encouragement, thoughts, and support of many individuals, including Dr. Erik DeBenedictis formerly of Bell Labs.; Prof. C.V. Ramamoorthy of the University of California, Berkeley; Prof. Anil Nerode of Cornell University; Prof. Larry Ho of Harvard University; Prof. Kinchi Mori of the Tokyo Institute of Technology; Dr. Jagdish Chandra formerly of the U.S. Army Research Office; Louis Lome of BMDO; Dr. David Hislop of the U.S. Army Research Office; Dr. Gottfried Luderer, emeritus professor at Arizona State University; Dr. Seong-Soon Joo, visiting research scientist in the Networking and Distributed Algorithms Laboratory at Arizona State University, 1996-1997 and 1999-2000, and currently department head at ETRI, Korea; Dr. Al Aho, vice president of Bell Labs (Lucent Technologies); Bernie Gallois, former dean of the School of Engineering at Stevens Institute of Technology; Dr. Frank Fernandez, former director of DARPA and currently director of institute initiatives at Stevens Institute of Technology; Prof. Stu Tewksbury of the ECE Department at Stevens Institute of Technology; Prof. Domenico Ferrari formerly of the University of California, Berkeley; Prof. Bernie Zeigler of the University of Arizona; Dr. Norm Sorensen, principal director of the Space Technology Directorate at the Aerospace Corporation; Colin Gillis, Eugene Kim, Arthur Chai, Raj Iyer, M.D., Mariano

Fernandez, Pierre Sorel, Peter Lee, Dr. Joanne Law, Tom Morrow, Noppanunt Utamaphethai, Kwun Han, Raymond Kuo, and Anish Bhimani, all former undergraduate researchers at Brown University; Drs. Tony Lee, Ling-Rong Chen, and Peter Walker, former doctoral and postdoctoral researchers at Brown University; Drs. Qutaiba Razouqi, Jerry Schumacher, P. Seshasayi, and Ricardo Citro, former doctoral researchers at Arizona State University; Peter Heck, former Ph.D. advisee at Arizona State University; and Subhendu Ghosh, currently Ph.D. advisee at Stevens. The author thanks all of the anonymous referees of the archival papers on which this book stands for their thoughtful comments and constructive criticism. The author is indebted to the U.S. Army Research Office and the BMDO Office for encouraging and supporting the research that underlies this book. For help with transforming a mental image of the book cover into an actual cover page, my sincere thanks to my son, Abhinav Ghosh. For their continued encouragement and support, Wayne Yuhasz and Wayne Wheeler of Springer Verlag deserve my utmost gratitude. Last, my sincere thanks to the entire production staff at Springer, especially Lesley Poliner.

Contents

Foreword	vii
Preface	xi
1 Introduction	1
1.1 What Are Networked Information Technology Systems?	1
1.2 What Are Asynchronous, Distributed Decision-Making Algorithms?	1
1.3 The Current Focus on Synchronous, Distributed Algorithms, and Why	4
1.4 Why Are Most Natural and Physical Processes Asynchronous?	6
1.5 The Importance of ADDM Algorithm-Controlled NIT Systems	7
1.6 Review of the Distributed Systems Literature	10
2 The Nature, Characteristics, and Implications of ADDM Algorithms	15
2.1 Nature of ADDM Algorithms	15
2.2 Fundamental Characteristics of ADDM Algorithms	16
2.2.1 Entities in ADDM Algorithms	16
2.2.2 Asynchronous Nature of the Entities	18
2.2.3 Concurrency in the Entities	19
2.2.4 Communication between the Entities	22
2.2.5 Proof of Correctness of ADDM Algorithms	23
2.2.6 Robustness of ADDM Algorithms	25
2.2.7 Performance of ADDM Algorithms	25
2.2.8 Stability of ADDM Algorithms	27
2.3 Existence and Synthesis of ADDM Algorithms for Real-World Problems	27

3 Case Studies: Synthesis of ADDM Algorithms for Real-World Systems 33

3.1 Algorithm for Accurate Execution of VHDL Descriptions 33

3.1.1 Introduction: Review of the Literature 33

3.1.2 The P^2EDAS Approach 38

3.1.2.1 The Role of the Event Prediction Network in P^2EDAS 39

3.1.2.2 Synthesis of the Event Prediction Network 42

3.1.2.3 The P^2EDAS Algorithm 46

3.1.3 An Example to Illustrate P^2EDAS 49

3.1.4 Implementation and Performance Analysis of P^2EDAS 59

3.1.4.1 Performance Results for Shared Memory Multiprocessor Implementation 59

3.1.4.2 Performance Results for Loosely Coupled Parallel Processor Implementation 66

3.1.4.2.1 DLX Computer Architecture 66

3.1.4.2.2 Random State Generator 67

3.1.4.3 Performance Analysis 68

3.2 Decentralized Algorithm for Railway Networks with Soft Reservation 70

3.2.1 Introduction 70

3.2.2 The RYNSORD Approach 71

3.2.3 Modeling RYNSORD on an Accurate, Realistic, Parallel Processing Testbed 78

3.2.4 Implementation Issues 86

3.2.5 Simulation Data and Performance Analysis 87

3.3 A Novel Architecture for Real-Time Payment Processing 101

3.3.1 Introduction 102

3.3.2 The NOVAHID Approach 105

3.3.2.1 The NOVAHID Algorithm 109

3.3.2.1.1 Execution of International Transactions 112

3.3.2.2 Modeling NOVAHID on a Loosely Coupled Parallel Processor System 114

3.3.3 Modeling in NOVAHID 116

3.3.4 Implementation and Performance of NOVAHID 119

3.3.4.1 CPU Time 121

3.3.4.2 Total Time Required for International Transactions 123

3.3.5 Limitations of NOVAHID 128

3.4 Decentralized Decision-Making in Military Command and Control 128

3.4.1 Introduction: History and Structure of Military Command and Control 134

3.4.2	A Novel Approach to Decentralized Command and Control	139
3.4.2.1	The Asynchronous, Decentralized Algorithm ..	139
3.4.2.2	Modeling the Decentralized C^3 Algorithm on an Accurate, Parallel Processing Testbed	141
3.4.3	Implementation Issues	152
3.4.4	Performance Analysis	152
3.4.4.1	Scenario 1	155
3.4.4.2	Scenario 2	163
3.4.4.3	Observations and Limitations	164
3.5	Distributed Circuit-Partitioning-Based Algorithm for Fault Simulation	165
3.5.1	Introduction	165
3.5.2	An Asynchronous, Distributed Algorithm for Fault Simulation	170
3.5.2.1	Distributed Fault Simulation of Combinational Subcircuits	171
3.5.2.2	Asynchronous, Distributed Fault Simulation of Sequential Subcircuits	173
3.5.2.2.1	Intuitive Overview	173
3.5.2.2.2	The NODIFS Algorithm for Sequential Subcircuits	175
3.5.3	Implementation Issues	187
3.5.4	Performance of NODIFS	189
3.6	Distributed Approach to Real-Time Payment Processing	191
3.6.1	Introduction	195
3.6.2	A Centralized, Uniprocessor-Based Algorithm for Real-Time Payment Processing	200
3.6.3	A Novel, Distributed Approach to Payment Processing ..	202
3.6.3.1	Execution of Events in the Pseudo-Transaction Algorithm	203
3.6.3.2	A Model of the Distributed Banking Network ..	205
3.6.3.3	A Distributed Architecture for the Simulation ..	206
3.6.3.3.1	Initialization from Network Configuration Files	207
3.6.3.3.1.1	Software Operating Characteristics of the Node	207
3.6.3.3.1.2	Simulation Characteristics of the Node	207
3.6.3.3.1.3	Characteristics of the Current Network	207
3.6.3.3.2	Connection Mechanism at Initialization	208

- 3.6.3.3.3 Assertion of Transaction Packets into the Simulation 209
- 3.6.3.3.4 The Notion of Time in the Simulation 209
- 3.6.4 Architecture of the Banking Simulation Program 209
 - 3.6.4.1 The Node Program 211
 - 3.6.4.1.1 Buffering and Processing of Input Packets 212
 - 3.6.4.1.2 Switching Packets through the Node 212
 - 3.6.4.1.3 Transmission and Buffering of Outgoing Packets 212
 - 3.6.4.2 Limitations of the Proposed Approach 213
- 3.6.5 Implementation Issues 213
- 3.6.6 Performance of Banking Network Simulation 214
 - 3.6.6.1 Experimental Banking Networks 214
 - 3.6.6.2 Results 215
- 3.7 Hierarchical, Distributed, Dynamic Inventory Management . . . 224
 - 3.7.1 Introduction 225
 - 3.7.2 Review of the Traditional Inventory Management Model 229
 - 3.7.3 The HDDI Approach 232
 - 3.7.4 Simulation Results and Performance Analysis of HDDI . 237
 - 3.7.4.1 Overview of HDDI's Performance Behavior . . . 240
 - 3.7.4.2 Performance Evaluation of HDDI Relative to Its Parameters for the Inventory Management Network 244
 - 3.7.4.2.1 The Parameter r_c , Ratio of Emergency to Regular Reorder Cost 244
 - 3.7.4.2.2 The Parameter r_p , Ratio of Regular to Emergency Order Size . . . 245
 - 3.7.4.2.3 The Parameter h_b , Ratio of Holding to Backorder Cost 247
 - 3.7.4.2.4 The Parameter r_l , Ratio of Normal to Emergency Lead Time 248
 - 3.7.4.2.5 The Parameter d_p , Daily Profit . . . 249
 - 3.7.4.2.6 Performance Comparison of Four Scenarios 249
 - 3.7.4.3 Performance Analysis for Different Inventory Management Networks 252
- 4 Debugging Network-Centric Systems 257**
 - 4.1 The Difficulty in Debugging NIT Systems 257
 - 4.1.1 Principal Reasons Underlying Debugging Difficulties . . . 259
 - 4.1.2 A Representative Experience in Debugging NIT Systems 259
 - 4.2 Visual Aids in Debugging: DIVIDE 261
 - 4.2.1 Introduction 261
 - 4.2.2 Visual Display of Parameters 261

4.2.2.1	Issues in User Interface Design	263
4.2.2.1.1	Movement Controls	263
4.2.3	Issues in the Design of the Graphical Display	266
4.2.3.1	Network Display	266
4.2.3.2	Buffer Graphs	266
4.2.3.3	Issues in Layout Design of the Graphical Images of the Concurrent Processes	267
4.2.3.4	Dynamic Scaling	268
4.2.3.5	Distributing the Visual Images of the Concurrent Processes	269
4.2.4	Implementation Details	271
4.2.5	Performance of DIVIDE	273
5	Proofs of Correctness of ADDM Algorithms	275
5.1	The Need for Proofs of Correctness of ADDM Algorithms	275
5.2	The Nature of the Proofs of Correctness of ADDM Algorithms	276
5.3	Proof of Correctness of the NODIFS Algorithm	277
5.3.1	Combinational Subcircuits	277
5.3.2	Sequential Subcircuits	279
5.3.2.1	Termination of Fault Simulation	280
5.4	Proof of Correctness of P^2EDAS	281
5.4.1	Execution of Events in the Correct Order: Monotonicity in W_h Values	281
5.4.2	Proof of Freedom from Deadlock	286
5.4.3	Termination of Simulation	287
6	A Mathematical Framework for Synthesizing ADDM Algorithms	289
6.1	Review of the Current Literature	289
6.2	A Military Command, Control, and Communication Problem	290
6.2.1	Component Vectors of the Entities	291
6.2.1.1	Movement State Vector	291
6.2.1.2	Sensor State Vector	291
6.2.1.3	Target State Vector	291
6.2.2	State Vectors of the Entities	292
6.2.2.1	Self State Vector	292
6.2.2.2	Friend State Vector	292
6.2.2.3	Enemy State Vector	292
6.2.3	Complete Internal State Vector of an Entity	293
6.2.4	Decision Vectors of Entities	293
6.2.4.1	Movement Decision	293
6.2.4.2	Sensor Management	293
6.2.4.3	Target Management	294
6.3	Modeling the C^3 Problem under the Centralized Paradigm in MFAD	294

- 6.3.1 Synthesizing Decision Vectors through Cost Functions . 294
 - 6.3.1.1 Cost Function Corresponding to the Movement Decision 295
 - 6.3.1.2 Cost Function Corresponding to the Target Decision 295
 - 6.3.1.3 Cost Function Corresponding to the Sensor Decision 296
 - 6.3.1.4 Centralized Lagrangian 296
- 6.4 Synthesizing Distributed Lagrangians for the C^3 Problem in MFAD 297
- 6.5 A Representative Battlefield Scenario and the Simulation Architecture 298
- 6.6 Performance Analysis of the Centralized and Decentralized Paradigms 302
- 7 Performance Analysis of ADDM Algorithms 311**
 - 7.1 Review of the Current Literature in Performance Analysis . . . 311
 - 7.2 The Traditional Performance Metric: Speedup and Its Limitations 313
 - 7.3 A Fundamental Framework for Evaluating Performance of NIT Systems 319
- 8 Stability of ADDM Algorithms 327**
 - 8.1 Review of the Current Literature in Stability of Distributed Systems 327
 - 8.1.1 Bounded-Input Bounded-Output Stability 327
 - 8.1.2 Marginal Stability 327
 - 8.1.3 Asymptotic Stability 328
 - 8.1.4 Self-Stabilization 328
 - 8.1.5 Stable Properties 329
 - 8.1.6 Robustness 330
 - 8.1.7 Fault-Tolerance 330
 - 8.1.8 Quality of Service 331
 - 8.2 Formal Definition of Stability in NIT Systems 331
 - 8.3 Stability Analysis, a Decentralized Command and Control Problem 334
 - 8.3.1 Modeling and Assumptions for Stability Analysis 334
 - 8.3.2 Steady-State Analysis 334
 - 8.3.3 Input Stability Analysis 335
 - 8.3.4 Perturbations to System Characteristics and Stability Analysis 337
 - 8.3.5 Summary of Stability Analysis of MFAD 337
 - 8.4 Stability Analysis of RYNSORD 339
 - 8.4.1 Modeling and Assumptions for Stability Analysis 339
 - 8.4.2 Steady-State Analysis 340

8.4.3	Perturbations to the Input Rate and Stability Analysis .	341
8.4.4	Perturbations to System Characteristics and Stability *Analysis	343
8.4.4.1	Perturbations to Interstation and Train-to-Station Communication	344
8.4.4.2	Perturbations Relative to the Track Segments .	346
8.4.5	Summary of Stability Analysis of RYNSORD	347
9	The Need and Art of Interdisciplinary Thinking	349
9.1	The Notion of Computational Intelligence	349
9.2	Reflection as a Catalyst in Triggering Creativity	351
9.2.1	Introduction	352
9.2.2	Reflection as a Catalyst in Triggering Creativity	354
9.2.2.1	Scientific Discoveries and Engineering Inventions	354
9.2.2.2	Creative Flashes in Ordinary Activities	357
9.2.2.3	Nature's Use of Reflection	358
9.2.3	Using Reflection to Trigger Creativity at Will	361
10	The Future of ADDM Algorithms	371
	References	377
	Index	391
	About the Author	401

List of Tables

3.1	Major Blocks in the DLX Architecture	66
3.2	Performance Results for DLX Architecture in P^2EDAS	67
3.3	Performance Results for a Synthetically Loaded Random State Generator	68
3.4	Input Train Traffic Parameters	89
3.5	Comparative Impact of Lookahead on Performance Parameters in RYNSORD under Low Input Traffic Density	99
3.6	Comparative Impact of Lookahead on Performance Parameters in RYNSORD under Medium Input Traffic Density	99
3.7	Control Packets for Communication between Control Node and Individual Nodes	145
3.8	Packets for Individual–Node–to–Individual–Node Communications	149
3.9	Packets to Facilitate Orders from Platoon and Company Leaders . .	149
3.10	Summary of Performance Measures for Scenario 1	162
3.11	Summary of Performance Measures for Scenario 2	164
3.12	Simulation Execution Times as a Function of Input Transactions and Network Size	218
3.13	Transaction Propagation Measures as a Function of Transaction Volume for 20-Node Banking Network	222
3.14	Cost Function Evaluations by R_1 s during Its Decision Making	236
3.15	Key Parameters in HDDI	238
6.1	Generic Characteristics of Each Entity	301
7.1	Summary of Performance Measures for Decentralized Military Command and Control during Deliberate Engagement	320
7.2	Summary of Performance Measures for Decentralized Military Command and Control during Withdrawal in the Face of Overwhelming Enemy Superiority	320

- 8.1 Average Sensor Decision Error Values Corresponding to Sensor Perturbations 335
- 8.2 Average Movement Decision Error Values Corresponding to Sensor Perturbations 337
- 8.3 Average Sensor Decision Error Values Corresponding to Communication Perturbations 338
- 8.4 Average Movement Decision Error Values Corresponding to Communication Perturbations 338
- 8.5 Input Traffic Parameters for Steady-State Analysis of RYNSORD . 341
- 8.6 Perturbations to Input Rate and System Characteristics 342
- 8.7 Performance Results for Communication Perturbations 346
- 8.8 Performance Results for Track Perturbations 347

List of Figures

2.1	The Concept of Universal Time	19
2.2	Understanding Concurrency in Hardware	20
2.3	Performance Scalability Relative to Increasing System Size	26
2.4	A Practical ADDM Algorithm Design from WWII	30
3.1	A Behavior Model of a Digital Component	39
3.2	Event Prediction Network	41
3.3	An Example Circuit to Illustrate the Role of the Event Prediction Network	42
3.4	Reducing a Cyclic Directed Graph to an Acyclic Directed Graph .	43
3.5	Event Prediction Network for the Cyclic Circuit in Figure 3.4(a) .	44
3.6	Digital Circuit with Cyclic Subcircuits	44
3.7	Event Prediction Network for Figure 3.6	45
3.8	Digital Circuit with Cyclic and Acyclic Subcircuits	45
3.9	Event Prediction Network for Figure 3.8	45
3.10	Operations of a Simulation Model X	50
3.11	Operations of a Head (X_h) or Tail (X_t) Pseudo Component	51
3.12	An Example Sequential Subcircuit	52
3.13	Event Prediction Network Corresponding to Figure 3.12	52
3.14	Step 1. Execution of Pseudo Components	53
3.15	Step 2. Execution of Models A and B Using New Window of Temporal Independence Values	54
3.16	Step 3. State of Subcircuit Following Execution of Models A and B	55
3.17	Step 4. Execution of the Event Prediction Network	56
3.18	State of Subcircuit with New Window of Temporal Independence Values	57
3.19	Step 5. State of Subcircuit Following Execution of Models A and B	57
3.20	Step 6. Execution of the Event Prediction Network	58

3.21	Step 7. State of the Subcircuit Following Execution of Models <i>A</i> and <i>B</i>	59
3.22	Step 8. Execution of the Event Prediction Network	60
3.23	Step 9. State of the Subcircuit Following Execution of Model <i>B</i> . .	61
3.24	Step 10. Execution of the Event Prediction Network	62
3.25	Step 11. Output Waveform Generated by Subcircuit Execution . .	63
3.26	Speedup Plot for Multiplier Circuits	64
3.27	Speedup Plot for Ripple Counters	64
3.28	Speedup Plots for RAM Circuits	65
3.29	Speedup Plots for SADD and BAR	65
3.30	DLX Computer Architecture	67
3.31	An Example Railway Network in RYNSORD	73
3.32	Computation of Primary and Secondary Paths in RYNSORD (a) for Train <i>Ta</i> and (b) for Train <i>Tb</i>	75
3.33	Initiation and Progress of Reservation Processing	79
3.34	Functionality of a Station Node in Pseudocode	81
3.35	Functionality of the Special Synchronizing Node	81
3.36	Screen Shot of the Graphical Interface	84
3.37	Screen Shot of an Actual Simulation Run	85
3.38	A Representative Railway Network: A 50-station subset of the Eastern United States Railroad Network	88
3.39	(a) Average over All Trains of (Actual Travel Time of a Train Minus Its Ideal Travel Time), as a Function of the Lookahead, and (b) Percentage of Trains Reaching Their Destinations within the Maximum Simulation Time	90
3.40	Distribution of Actual Number of Hops for Trains in RYNSORD vs. Approach <i>B</i> vs. Ideal Distribution	91
3.41	Distribution of Actual Number of Hops for Trains in RYNSORD vs. Ideal Distribution under (a) Low Input Train Density, (b) Medium Input Train Density, and (c) High Input Train Density . .	93
3.42	Average of (Actual Hops for Each Train Minus Its Ideal Hops) Over All Trains, as a Function of Lookahead	94
3.43	Actual Travel Time of a Train in RYNSORD Minus Ideal Travel Time for Lookahead 4 as a Function of Assertion Time of the Train in, (a) Low Input Traffic Density, (b) Medium Input Traffic Density, and (c) High Input Traffic Density	95
3.44	Track Utilization for Low Input Traffic Density and Lookahead 6 in, (a) Approach <i>B</i> , and (b) RYNSORD	96
3.45	Track Utilization in RYNSORD for Medium Input Traffic Density in (a) Lookahead 2, and (b) Lookahead 4	97
3.46	Distribution of Reservations Processed by Stations in RYNSORD for Medium Input Traffic Density as a Function of Stations in (a) Lookahead 2, and (b) Lookahead 4	97

3.47	Distribution of Computation among the Trains in RYNSORD for Medium Input Traffic Density for (a) Lookahead 2 and (b) Lookahead 4	98
3.48	Maximum Communications Rate (bytes/minute) of Interstation Links	100
3.49	A Representative NOVAHID Model Consisting of Three Groups, Each with Two Group-Nodes	108
3.50	NOVADIB Algorithm for a Group Network (or Top-Level Network) with Two Group-Nodes (or Int-Nodes)	110
3.51	Representation of the Top-Level Network in NOVADIB	112
3.52	Representation of the United States Group-Network in NOVADIB	112
3.53	Transactions Asserted at the NOVAHID Model	117
3.54	Step 0	120
3.55	Step 1	121
3.56	Step 2	122
3.57	Step 3	123
3.58	Step 4	124
3.59	Step 5	125
3.60	Step 6	126
3.61	Step 7	127
3.62	Step 8	128
3.63	Step 9	129
3.64	CPU Time (millisec) required by All Transactions as a Function of the Number and composition of Transactions: (a) Model 2.3, (b) Model 3.3, and (c) Model 2.4	130
3.65	Total Simulation Time (sec) Required for Processing the Marked International Transactions in the 3.3 Model (sleep factor 0)	131
3.66	Total Simulation Time (sec) Required for Processing the Marked International Transactions as a Function of the Volume of Transactions, and Sleep Factor Values in (a) Model 2.3, (b) Model 3.3, and (c) Model 2.4	132
3.67	Total Simulation time (sec) Required for Processing the Marked International Transactions as a Function of the Volume of Transactions and for a Sleep Factor Value of 5 sec in (a) Model 2.3, (b) Model 3.3, and (c) Model 2.4	133
3.68	A Generic Individual Node	142
3.69	Behavior of the Generic Individual Node	142
3.70	Communications in Centralized and Decentralized Tank Platoons	143
3.71	Network Connectivity of a Battlefield at the Company Level	144
3.72	Flow of Communications and Decisions at the Company Level with (a) Centralized C^3 and (b) Decentralized C^3	148
3.73	Behavior of Decentralized Combat Units, Including Tanks, Platoon Leaders, and Company Leader	150
3.74	Behavior of Decentralized Artillery Unit	151
3.75	Time Lag between Enemy Unit Sighting and Artillery Fire	152

3.76 Progress of Simulation: Screen Shot at Timestamp 2 (Blue, Decentralized Force on the Left and Red, Centralized Force on the Right) 157

3.77 Progress of Simulation: Screen Shot at Timestamp 35 (Blue, Decentralized Force on the Left and Red, Centralized Force on the Right) 158

3.78 Progress of Simulation: Screen Shot at Timestamp 98 (Blue, Decentralized Force on the Left, and Red, Centralized Force on the Right) 159

3.79 Progress of Simulation: Screen Shot at Timestamp 228 (Blue, Decentralized Force on the Left, and Red, Centralized Force on the Right) 160

3.80 Progress of Simulation: Screen Shot at Timestamp 423 (Blue, Decentralized Force on the Left, and Red, Centralized Force on the Right) 161

3.81 The Nature of Influence of Company Leader Processing Delay on Battlefield Performance 163

3.82 Fault Simulation of a Simple Circuit 166

3.83 A Combinational Component 171

3.84 NODIFS Algorithm for Combinational Subcircuits 174

3.85 Reducing a Cyclic Directed to an Acyclic Directed Graph 176

3.86 Data-Flow Network Constructed for Cyclic Circuit in Figure 3.85(a) 176

3.87 The NODIFS Approach 178

3.88 Operations of a Model X 181

3.89 An Example Digital Design for NODIFS 182

3.90 Snapshots of the Fault Simulation Process of the Example Design in Figure 3.89 185

3.91 Another Example Design for Fault Simulation in NODIFS 187

3.92 Progress of Fault Simulation of the Example Design in Figure 3.91 188

3.93 Fault Simulation of Two Sequential Circuits in NODIFS: (a) A Sequential Circuit with Three Feedback Loops, and (b) Modified SN7482 Adder with Feedback 192

3.94 Performance Graphs for the Three-Loop Example Circuit in Figure 3.93(a) 193

3.95 Performance Graphs for the Modified SN7482 Adder with Feedback 193

3.96 Performance Graphs for the SN74251 Multiplexer 194

3.97 Performance Graphs for the SN7482 Adder 194

3.98 Performance Graphs for the 74181 ALU 195

3.99 A Partial Organizational Layout of the United States Federal Reserve System 197

3.100 A Centralized, Uniprocessor-Based Approach for Payment Processing 201

3.101 Algorithm for Determining the Execution of Events in PTA 203

3.102	An Example Distributed Banking Simulation	204
3.103	A Graphical Representation of a Banking Node	210
3.104	A 10-Node Banking Network	215
3.105	A 20-Node Banking Network	216
3.106	A 50-Node Banking Network	217
3.107	CPU Time (ms) per Transaction (Real) as a Function of Network Size	219
3.108	CPU Time (μ s) per Transaction (Real and Pseudo) as a Function of Network Size	220
3.109	Average Output Queue Size between Nodes 1 and 13 for a 20-Node Banking Network, as a Function of Transaction Assertion Time	221
3.110	Transaction Completion Times as a Function of the Transaction Volume	221
3.111	Transaction Completion Times as a Function of Increased Values of Transaction Volume	223
3.112	Completion Times for Transactions in a System with Priorities Corresponding to 70% Transaction Volume	224
3.113	Completion Times for Transactions in a System with Priorities Corresponding to 50% Transaction Volume	225
3.114	A Traditional Inventory Management Network	227
3.115	Traditional Inventory Model and the Proposed Emergency Replenishment	230
3.116	An Example Inventory Management System Consisting of Two Neighborhoods	237
3.117	A Representative Inventory Management System	241
3.118	Consumer Demand Functions for Retail Units in (a) a RI Neighborhood and (b) a MA Neighborhood.	242
3.119	Overview of HDDI's Performance Behavior: Inventory Levels for (a) Retail Unit #8 in the Massachusetts Neighborhood, and (b) All Warehouses	243
3.120	(a) Cumulative Cost Function as a Function of Simulation Time, for Different Values of r_c ; (b) Plot of Four Measures—Back orders, Demands, Refusals, and Number of Pieces Transferred— as a Function of Simulation Time, for $r_c = 3$ and $w_c = 4$; (c) Plot of Four Measures—Back orders, Demands, Refusals, and Number of Pieces Transferred—as a Function of Simulation Time, for $r_c = w_c = \infty$; (d) Variation of Cumulative Cost Function and Number of Pieces Transferred as a Function of r_c and w_c Parameter Values	246
3.121	(a) Normalized, Cumulative Cost Function Value as a Function of the Ratio, r_p ; (b) Cumulative Cost Function Value and Number of Items Transferred under Emergency Replenishment as a Function of r_p	247

3.122 (a) Normalized, Cumulative Cost Function Value as a Function of the Parameter, h_b ; (b) Plot of Four Measures—Back orders, Demands, Refusals, and Number of Pieces Transferred—as a Function of Simulation Time, for Low Back order Cost Given by $h_b = 1.0$ 248

3.123 Plot of Four Measures—Back orders, Demands, Refusals, and Number of Pieces Transferred—as a Function of Simulation Time, for (a) $r_l = 3$ and $w_l = 5$, (b) $r_l = 1$ and $w_l = 3$ 249

3.124 Cumulative Cost Function Value as a Function of the Daily Profit per Item, d_p 250

3.125 (a) Cumulative Cost Function Values for Scenarios 1 through 4; (b) Plot of Back orders, Demands, Refusals, and Number of Pieces Transferred, as a Function of Simulation Time, for Back orders and Random Demand; (c) Plot of Back orders, Demands, Refusals, and Number of Pieces Transferred, as a Function of Simulation Time, for Back orders and Fixed Demand; and (d) Plot of Back orders, Demands, Refusals, and Number of Pieces Transferred, as a Function of Simulation Time, for Fixed Demand and Absence of Back orders 251

3.126 Four Representative Inventory Management Systems: (a) Network 1 with 5 Entities; (b) Network 2 with 8 Entities; (c) Network 3 with 12 Entities; and (d) Network 4 with 16 Entities . . . 253

3.127 (a) Wall Clock and Normalized Execution Times (milliseconds) for Different Networks, on Linux-Based Pentium Testbed; (b) Execution Time (milliseconds) of Network 2 and Network 4 on the Collection of 17 Sun Sparc Machines Configured as a Loosely Coupled Parallel Processor; and (c) Total Measured Execution Times (milliseconds) for Networks 2 through 4, in HDDI and Projected Execution Time for the Traditional, Centralized Approach 254

3.128 Normalized, Cumulative, Cost Function Value as a Function of Network Size 256

4.1 Computing the Average of a Parameter over a User-Specified Time Period 263

4.2 An Actual Screen Display in DIVIDE. 264

4.3 Computing the Maximum Number of Nodes for a Circular Layout 268

4.4 Architecture of DIVIDE 273

5.1 An Example Circuit to Illustrate the Monotonicity in the W_h Values 282

5.2 Event Prediction Network for a Circuit in Figure 5.1 283

6.1 Sensor Arc and Sensor Axis for an Entity 292

6.2	Interconnection under the Centralized and Decentralized Paradigms	299
6.3	Interconnection Network for the Blue and Red Forces in the Simulation	300
6.4	A Screen Shot of the Post-mortem Replay Interface	302
6.5	Number of Enemy Units Destroyed under the Centralized and Decentralized Paradigms, as a Function of τ_c	304
6.6	Sensor Decision Vector Graph for an Entity under the Centralized and Distributed Paradigms and PGOD, as a Function of Time, for (a) $\tau_c = 1$ Timestep, (b) $\tau_c = 5$ Timesteps, and (c) $\tau_c = 9$ Timesteps.	305
6.7	Sensor Error Measures for an Entity for Centralized, Corrected Centralized, and Distributed Scenarios, as a Function of Time, for (a) $\tau_c = 1$ Timestep, (b) $\tau_c = 5$ Timesteps, and (c) $\tau_c = 9$ Timesteps.	306
6.8	Average Sensor Error over All Entities for Centralized, Corrected Centralized, and Distributed Scenarios, as a Function of τ_c	307
6.9	Average Movement Error over All Entities for Centralized, Corrected Centralized, and Distributed Scenarios, as a Function of τ_c	308
7.1	Graphs for (a) Speedup in Behavior-Level Simulation of Digital Systems, (b) Superlinear Speedup	315
7.2	Performance of DARYN, a Distributed Scheduling Approach for Railway Networks	318
7.3	Percentage Differential Travel Times for All Vehicles, i.e., Difference of Actual Travel Time of Each Vehicle Relative to Its Ideal Travel Time as a Fraction of the Ideal Travel Time, for (a) Density 3, (b) Density 6, and (c) Density 8.	321
7.4	Distribution of Actual Number of Hops for Trains in RYNSORD vs. Ideal Distribution	325
7.5	Sensor Decision Vector for an Entity under the Centralized and Distributed Paradigms and PGOD, as a Function of Time, for $\tau_c = 1$ Timestep	326
8.1	Average Sensor Error as a Function of Simulation Time, for Sensor Range Reduced to 33% and No Sensor Perturbation	336
8.2	Movement and Sensor Decision Errors as a Function of Simulation Time, Corresponding to Communication and Sensor Perturbations of Finite Durations	338
8.3	Error Criterion 1 for Each Train as a Function of the Assertion Time, for Input Rate = 0.125	342
8.4	Error Criterion 2 as a Function of Simulation Time, for Steady-State Input Rates	343

- 8.5 Error Criterion 2 as a Function of Simulation Time, for Different
Input Perturbations 344
- 9.1 Interdisciplinary Nature of Information Technology Systems 350
- 9.2 A Simple String Search Algorithm 366
- 9.3 First Improvement of the KMP Algorithm 367
- 9.4 Second Improvement of the KMP Algorithm 368
- 9.5 Discovering a Third Improvement over the Simple String Search
Technique 368